1969

Determination of direct trees by T-triangle method.

Willman, Carl Edward

Monterey, California. U.S. Naval Postgraduate School

DETERMINATION OF DIRECT TREES BY T-TRIANGLE METHOD

by

Carl Edward Willman

United States Naval Postgraduate School

THESIS

DETERMINATION OF DIRECT

TREES BY T-TRIANGLE METHOD

by

Carl Edward Willman

April 1969

DETERMINATION OF DIRECT TREES BY T-TRIANGLE METHOD

by

Carl Edward Willman
Lieutenant, United States Navy
B.S.E.E. University of Washington, 1961

Submitted in partial fulfillment of the
requirements for the degree of
MASTER OF SCIENCE IN ELECTRICAL ENGINEERING
from the
NAVAL POSTGRADUATE SCHOOL
April 1969

ABSTRACT

A method for determining all of the direct trees of an oriented graph is presented. This method, the T-triangle method, is suitable for either hand calculation or computer implementation. The method is simple, contains a minimal amount of steps, and generates all of the direct trees without any duplications.

TABLE OF CONTENTS

1. INTRODUCTION

Topological formulas have been developed in recent years for writing certain classes of network functions (mainly driving point and transfer admittances). The reason for the popularity of topological analysis is the relative simplicity of evaluating the network determinants and cofactors in comparison with the general conventional method. The topological method of analysis naturally led to digital computer implementation due to its time saving property.

The development of topological formulae was initially mainly concerned with passive networks without mutual inductances as discussed in (5) and led to the formulation of "tree products". The inclusion of active networks including mutual inductances led to such formulations as "complete-tree products" as discussed in (6) and "direct tree products" as discussed in (3) and (4).

All of these methods utilize the trees obtained from a given network graph and many investigators have developed methods for obtaining the trees of a network. How all of the "direct trees" of a given oriented graph could be found by using the basic T-triangle theory of Chan and Chang (1) was the basis of this thesis.

In section 2, some basic topological definitions are given and basic topological formulas are briefly discussed. Section 3 contains the T-triangle method of finding all of the trees and direct trees of a given oriented graph G. The main advantages of this method are simplicity, non-duplication of trees, and suitability for hand calculation and digital computer programming. An algorithm, proof of method, and examples are also included in section 3.

2. NETWORK TOPOLOGY

Network topology or network graph theory is a mathematical study of networks in connection with their non-metric geometrical properties by investigating the interconnections between the nodes and branches of the networks.

2.1 BASIC TOPOLOGY DEFINITIONS FOR NON-ORIENTED GRAPH

Given the passive electrical network shown in figure 2(a), an admittance graph Y of the network is formed in 2(b). There are 4 nodes and 5 branches in Y. If each branch is replaced by a line segment or an arc and the nodes remain unchanged, a topological graph G is formed in 2(c).

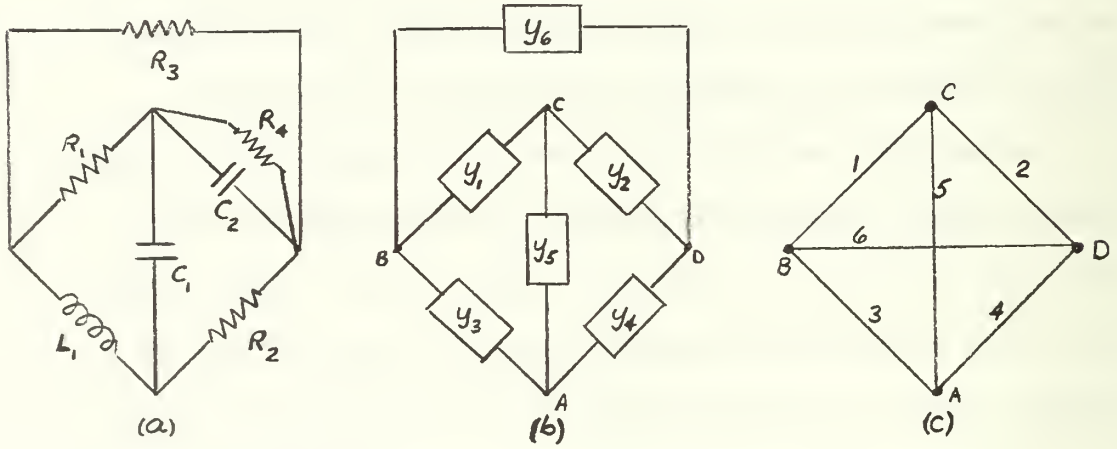


Figure 2-1

Following are some definitions of the graph G.

Graph - a topological representation of a network.

Edge or element - a line segment of G together with its end points.

Vertex (node) - an end point of an edge.

Degree of vertex - the number of edges incident to the vertex.

Subgraph - a subset of edges of the graph.

Circuit (Loop) - a closed path in a graph; a subgraph in which every vertex is of degree 2.

Tree - a connected subgraph which contains all of the nodes of the network but does not contain any closed path.

Cotree - the complement set of edges not contained in a tree.

Chords - the edges of a cotree.

2-tree - a subgraph of a tree formed by removing one of its edges.

3-tree - a subgraph of a 2-tree formed by removing one of its edges.

Tree admittance product - a product of the branch admittances of a tree T.

2-tree admittance product - a product of the branch admittances of a 2-tree.

Figure 2-2 illustrates some structures of circuits, trees, 2-trees, and cotrees of a graph G.

Some of the topological formulas are briefly stated below without proof or discussion. Reference is made to (5) or (6) or other works on network analysis for further discussion.

Node admittance determinant (Δ_n)

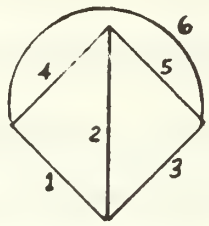
$$\Delta_n = \sum_{\text{all trees}} \text{tree admittance products of a network}$$

(Maxwells rule or formula)

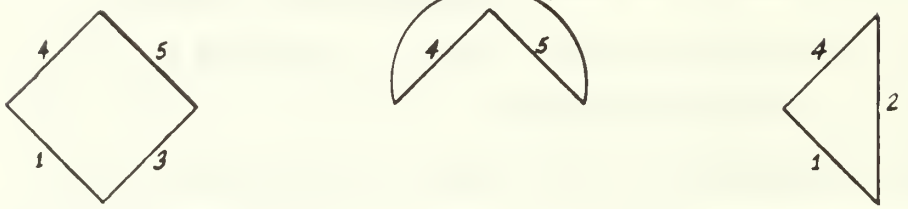
Loop impedance determinant (Δ_m)

$$\Delta_m = \sum_{\text{all trees}} \text{chord set impedance products of the network}$$

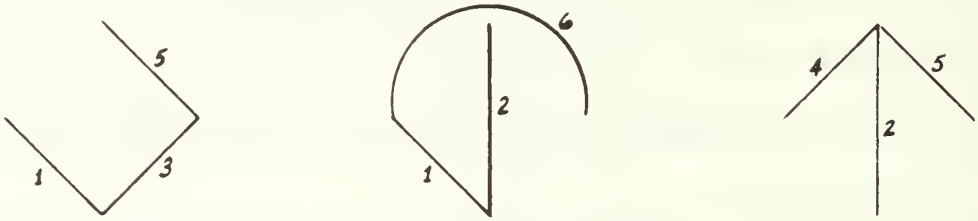
(Kirchoffs rule or formula)



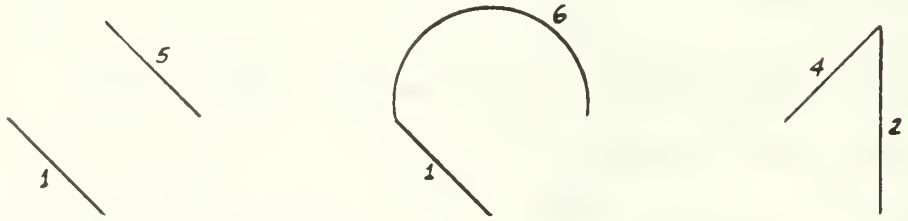
GRAPH "G"



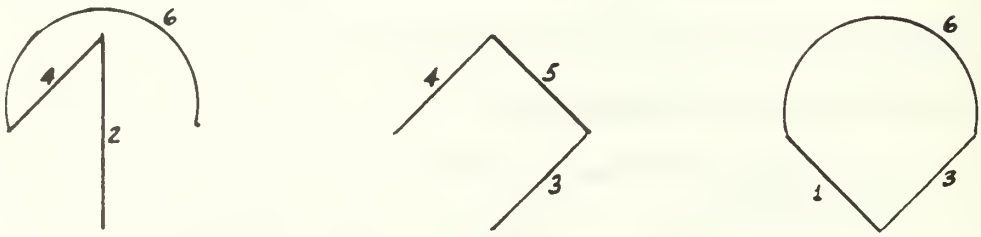
(a) Some circuits of G



(b) Some trees of G



(c) Some 2-trees of G



(d) Cotrees of part (b)

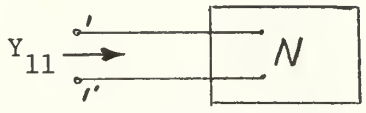
Figure 2-2

Cofactor of the node admittance matrix (Δ_{ij})

$\Delta_{ij} = \sum_{\text{all trees}}$ 2-tree (ij,r) admittance products of a network where each 2-tree is formed in two subgraphs; one subgraph contains nodes i,j and the other subgraph contains the reference node r.

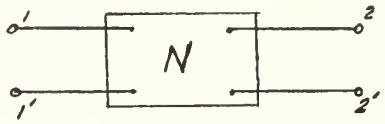
Driving point admittance (Y_{11})

$$Y_{11} = \frac{\Delta_n}{\Delta_{11}}$$



Transfer admittance (Y_{21})

$$Y_{21} = \frac{\Delta_n}{\Delta_{12} - \Delta_{12'}}$$



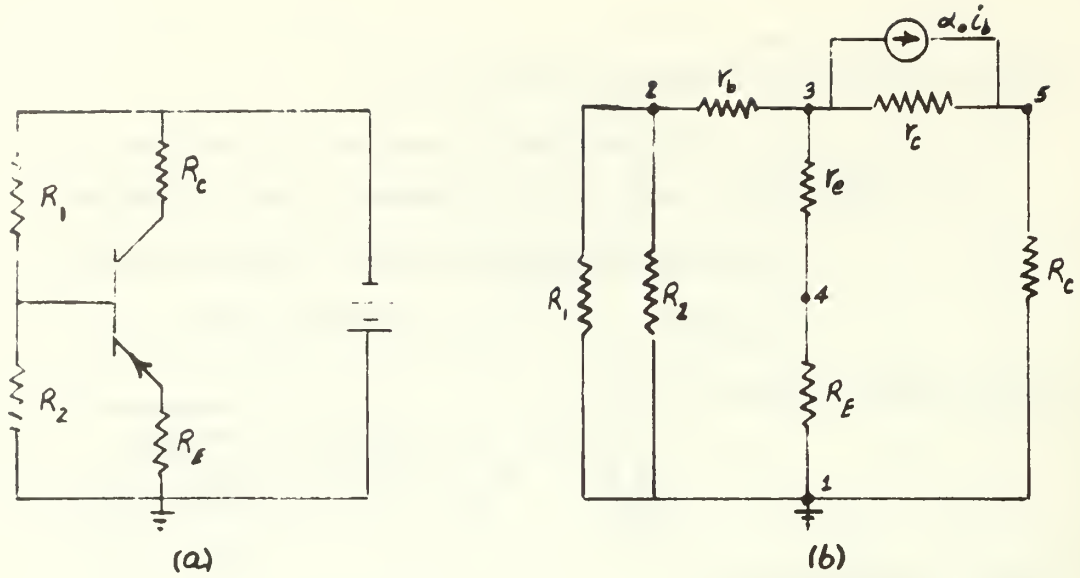
2.2 ORIENTED OR DIRECTED GRAPH

An oriented or directed graph is a graph of a network N that has positive current orientation assigned to the branches of the graph. By convention, a current arrow, corresponding to positive current flow, is superimposed on the graph branches. The oriented graph is necessary when working with active devices in networks. Figure 2-3 illustrates the generation of an oriented graph from a given transistor network.

All of the definitions given previously for a non-oriented graph pertain to the oriented graph with the inclusion of current sense given to the elements. Following are some extended definitions for the oriented graph.

Direct tree - a direct tree is a tree with all branches oriented toward the reference node.

Direct 2-tree - a subgraph of a direct tree formed by removing one of its edges.



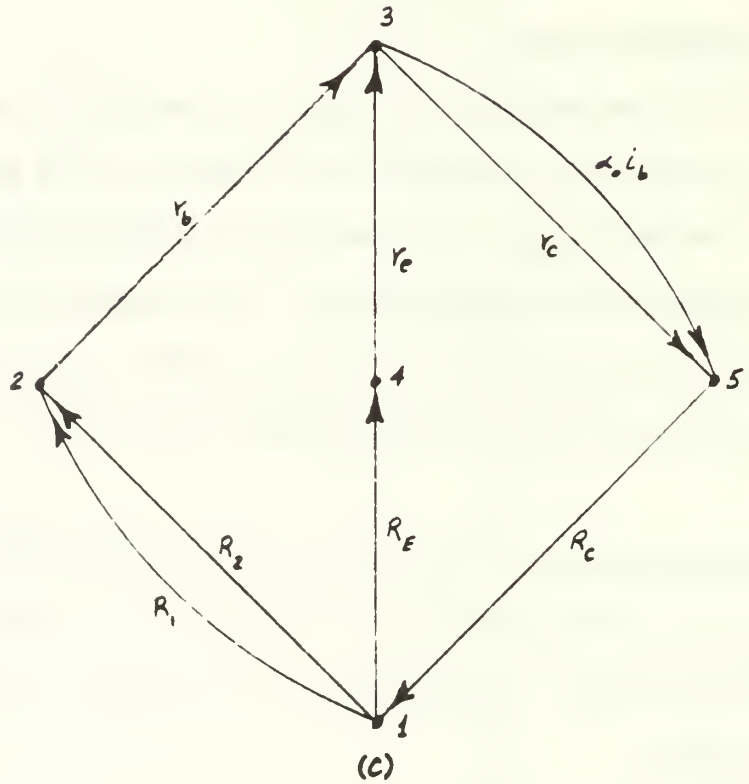


Figure 2-3 (a) transistor amplifier circuit
(b) Tee-equivalent mid-frequency circuit
(c) corresponding oriented graph

Direct tree admittance product - a product of the branch admittances of a direct tree.

Direct 2-tree admittance product - a product of the branch admittances of a direct 2-tree.

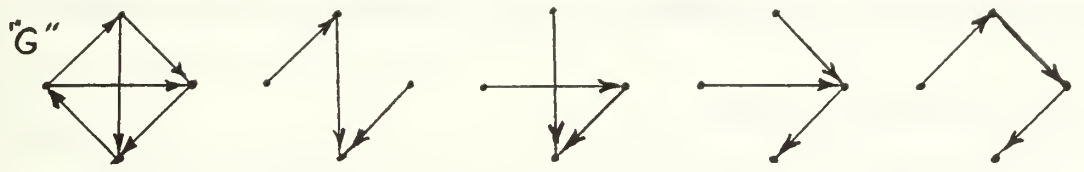


Figure 2-4 Direct trees of graph G

Chen in (4) develops topological formulaes for oriented graphs in relation to direct trees. Given below are the results obtained by Chen.

Node admittance determinant (Δ_n) of an oriented graph

$$\Delta_n = \sum_{\text{all direct trees}} \text{direct tree admittance products of a network.}$$

Cofactor of the node admittance matrix (Δ_{ij}) of an oriented graph. $\Delta_{ij} = \sum_{\text{all direct 2-trees}}$ direct 2-tree (ij,r) admittance products of a network where each direct 2-tree is formed in two subgraphs; one subgraph contains nodes i,j and the other subgraph contains the reference node r.

The determination of all direct trees of a network is therefore required in order to utilize the topological formulaes formulated by Chen. The following section develops a method for determining all of the direct trees of a network.

3. DIRECT TREES BY T-TRIANGLE METHOD

The direct tree finding method is based on obtaining the T-triangles from a given oriented graph. The method of obtaining T-triangles is presented in (1) and (2) and will be basically repeated here with the inclusion of modifications for direct trees.

3.1 NODAL CONNECTION MATRIX

The T-triangles are derived from the nodal connection matrix. This matrix displays the connecting elements between the oriented graph nodes.

A nodal connection matrix of an oriented graph is a rectangular array of which the element E_{ij} in the i^{th} row and j^{th} column of the connection matrix is given by

$$E_{ij} = \left\{ I_i \bigcap I_j \;\middle|\; \begin{array}{l} i = 1,2,3,\ldots. v \\ j = 1,2,3,\ldots. v \end{array} \right\}$$

where I_k is the set of all edges incident to the node k, v is the number of nodes in the graph, and the sense of E_{ij} is taken from i to j. Some properties and theorems of the connection matrix are listed below.

Property 3.1

Both the number of rows and columns of a connection matrix are v. The connection matrix is therefore a square matrix of order v.

Property 3.2

The sense of a connecting element, E_{AB} is positive if by going from node A to node B the current direction is from A to B.

Theorem 3.1

There are (v-1)! connection matrices to represent a given graph G.

Proof: With the exception of the reference node, which is predetermined, the numbering of the remaining nodes is strictly arbitrary and there are therefore (v-1) remaining nodes which have (v-1)! possible permutations.

Property 3.3

The connection matrix has no elements along the main diagonal and is formed of two triangular arrays which are symmetric except for opposite sense.

Figure 3-1 illustrates the generation of all possible connection matrices from a given oriented graph G.

3.2 T-TRIANGLES

The T-triangle method of generating direct trees will utilize the lower triangular array of the connection matrix.

A T-triangle of a directed graph is a triangular array of which the element t_{ij} in the i^{th} row and the j^{th} column of the T-triangle is given by

$$t_{ij} = \left\{ I_{i+1} \cap I_j \;\middle|\; \begin{array}{l} i = 1,2,3,\ldots\ v-1 \\ j \leq i \end{array} \right\}$$

where I_k is the set of all edges incident to node k, v is the number of nodes in the graph, and the sense of t_{ij} is taken from i to j.

Figure 3-2 illustrates the formation of both a connection matrix and the corresponding T-triangle from a given graph G.

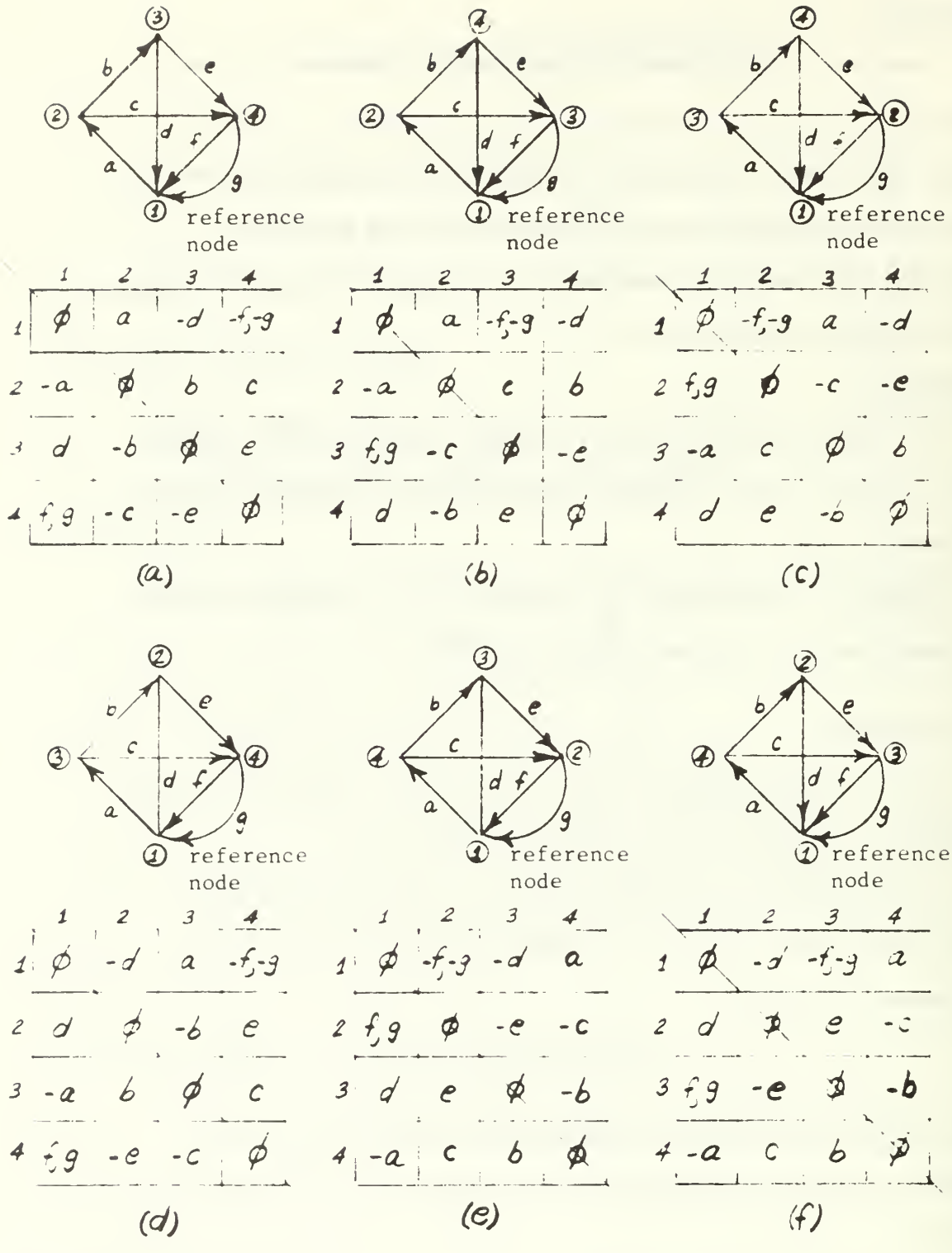
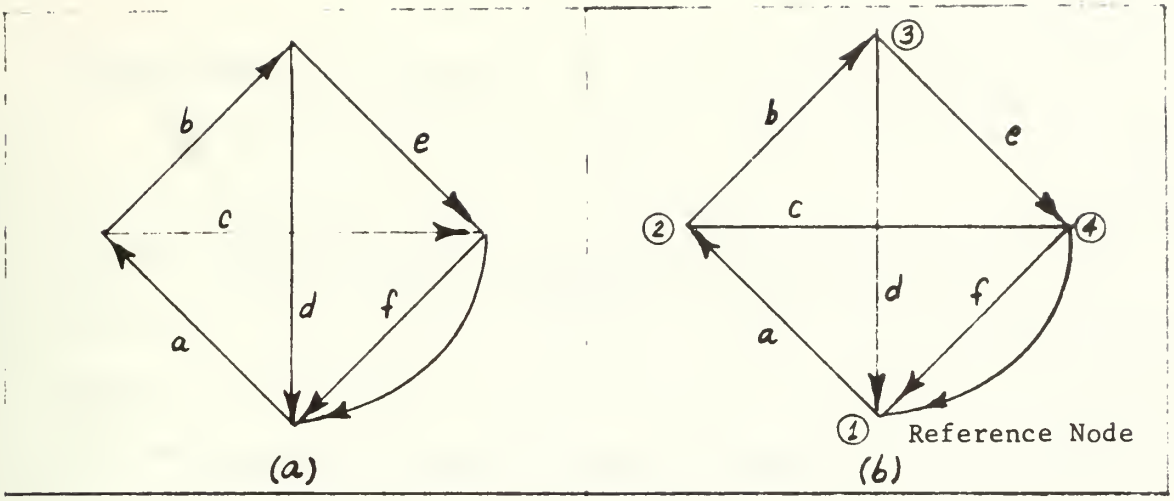
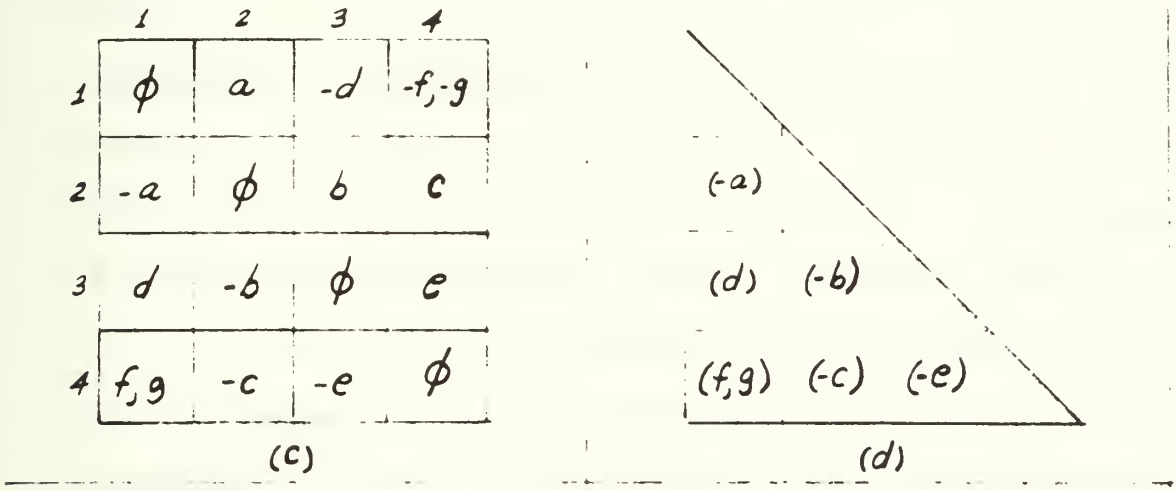


Figure 3-1 All Possible Connection Matrices For A Given Graph With Reference Node Specified





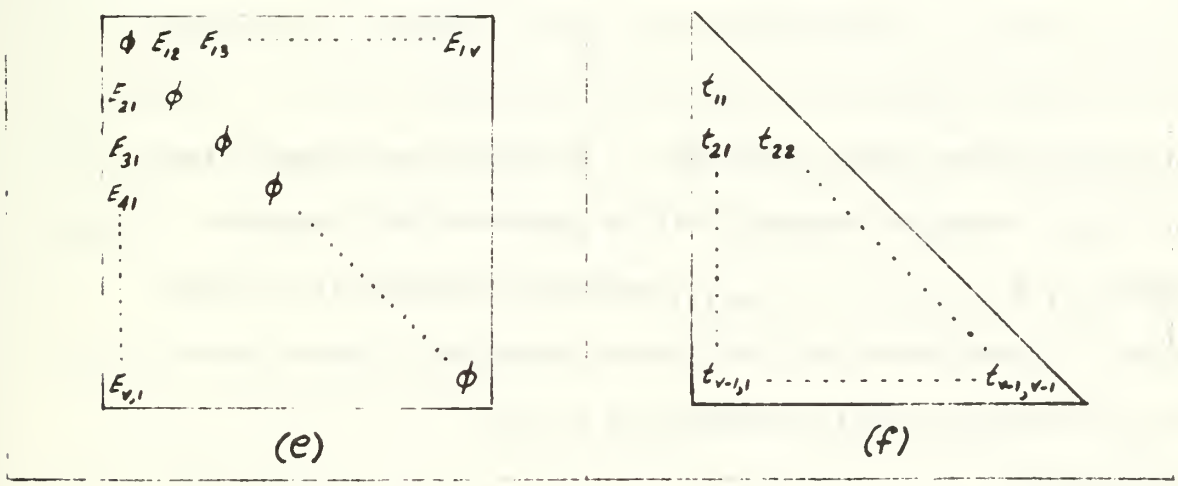


Figure 3-2 (a) Oriented Graph, (b) Graph with reference node selected, (c) Nodal Connection Matrix, (d) T-triangle, (e) and (f) are generalized forms of Connection Matrix and T-triangle respectively

Some properties and theorems of the T-triangle are given below.

Property 3.4

Both the number of rows and the number of columns of a T-triangle are (v-1).

Theorem 3.2

There are (v-1)! possible T-triangles formed from a given oriented graph.

Proof: This follows from the theorem that there are (v-1)! possible connection matrices.

Theorem 3.3

By taking one element (non-zero) from each row of a T-triangle, a tree of the graph G is formed.

Proof: This stems directly from the definition of a tree, "... a connected subgraph which contains all of the nodes of the network but does not contain any closed path." Element t_{11} is connected between nodes 2 and 1 and likewise element t_{ij} is connected between nodes (i+1) and j. By taking one element from each row a connected subgraph will be completed which contains edges t_{11}, t_{2j}, t_{3j}, ... $t_{(v-1)j}$ and which connects all of the nodes v. Since there are (v-1) edges connecting v nodes, there is no closed path and the subgraph is a tree.

Theorem 3.4

There are a maximum possible (v-1)! trees for each T-triangle having no parallel branches.

Proof: If there were no empty elements in the T-triangle there would be exactly (v-1)! combinations for the (v-1) rows.

Theorem 3.5

The directed or oriented trees are those trees obtained from the T-triangles that have all elements positive.

Proof: This follows directly from the definition of a directed tree, ".... a tree with all branches oriented toward the reference node." The t_{11} element of a direct tree must be positive because its sense is taken with respect to a reference node. The second element $t_{2j}|^{j=1,2}$ must also be positive as its sense is taken with either the reference node (j=1) or with node 2 (j=2) which connects to t_{11}. Likewise, the third element of the direct tree $t_{3j}|^{j=1,2,3}$ must also be positive as its sense is taken with either the reference node or the nodes connecting elements t_{11} or t_{2j}. By induction, all elements must therefore have a positive sense.

Theorem 3.6

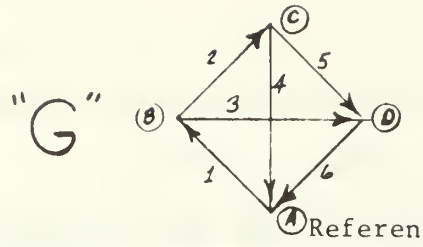
For a connected graph having no parallel branches, there are a maximum of (v-1)! trees for each triangle or a maximum total of $[(v-1)!]^2$ trees for the graph of which many are duplications.

Proof: This comes directly from theorems 3.2 and 3.4.

Figure 3-3, table 3-1, and figure 3-4 illustrate the generation of all trees of a given graph G.

3.3 GENERATION OF ALL CONNECTION MATRICES

The (v-1)! connection matrices can all be derived from performing elementary row and column transformations on a given connection matrix. Given below is such a procedure which will generate all of the (v-1)! connection matrices without duplication.



Permutations
ABCD
ABDC
ACBD
ACDB
ADCB
ADBC

	A	B	C	D
A	∅	1	-4	-6
B	-1	∅	2	3
C	4	-2	∅	5
D	6	-3	-5	∅

(a)

	A	B	D	C
A	∅	1	-6	-4
B	-1	∅	3	2
D	6	-3	∅	-5
C	4	-2	5	∅

(b)

	A	C	B	D
A	∅	-4	1	-6
C	4	∅	-2	5
B	-1	2	∅	3
D	6	-5	-3	∅

(c)

	A	C	D	B
A	∅	-4	-6	1
C	4	∅	5	-2
D	6	-5	∅	-3
B	-1	2	3	∅

(d)

	A	D	C	B
A	∅	-6	-4	1
D	6	∅	-5	-3
C	4	5	∅	-2
B	-1	3	2	∅

(e)

	A	D	B	C
A	∅	-6	1	-4
D	6	∅	-3	-5
B	-1	3	∅	2
C	4	5	-2	∅

(f)

Figure 3-3 All Possible Connection Matrices (T-triangles) For G

Connection Matrices (T-triangles)

Trees	(a)	(b)	(c)	(d)	(e)	(f)
1	-1,4,6	-1,5,4	4,-1,6	4,6,-1	6,4,-1	6,-1,4
2	-1,4,-3	-1,-3,4	4,-1,-3			
3	-1,4,-5		4,-1,-5			
4	-1,-2,6	-1,6,-2				6,-1,-2
5	-1,-2,-3	-1,-3,-2				
6	-1,-2,-5					
7		-1,6,5			6,5,-1	6,-1,5
8		-1,-3,5				
9			4,2,-3			
10			4,2,6	4,6,2	6,4,2	
11			4,2,-5	4,-5,2		
12				4,6,3	6,4,3	6,3,4
13				4,-5,3		
14					6,5,3	6,3,5
15					6,5,2	
16						6,3,-2

Table 3-1
Trees of Figure 3-3

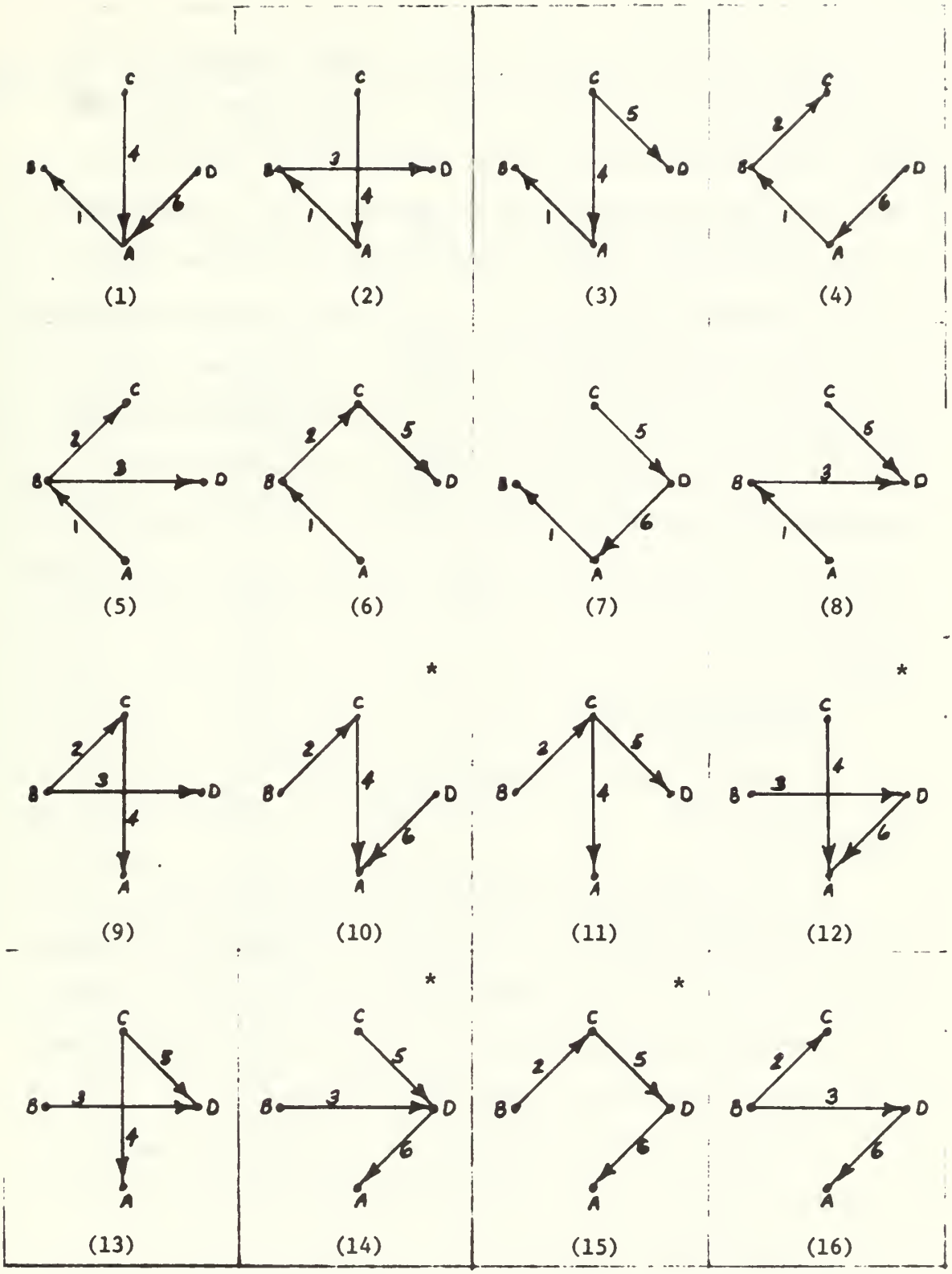


Figure 3-4 Trees of Table 3-1

(* indicates directed tree)

Given a connection matrix M, a new unique connection matrix M_2 can be generated by translating the second column out to the end column and by translating the second row down to the bottom row. The same process can then be repeated on M_2 to generate M_{22} (where the 2 in the subscript denotes the row and column number being translated and the amount of subscripts indicates the amount of times the translations have occurred). The process can then be repeated to form M_{222} from M_{22}. If these transformations are done exactly v-j times (j=2), column and row 1 will have all possible permutations with column and row 2. Further translations will result in duplications. Figure 3-5 illustrates the formation of M_2 from M_0.

Example 3-1

v=5 (M_0=12345), j=2

M_0=12345 $\rightarrow$ M_2=13452 $\rightarrow$ M_{22}=14523 $\rightarrow$ M_{222}= 15234 $\rightarrow$ M_{2222}=12345 a duplication.

If all of these new connection matrices M_0, M_2, M_{22}, ..., $M_{22...2}$ are now utilized by performing v-3 translations on the third column and row of each given matrix, thus forming v-3 new matrices from each given matrix, all possible permutations between the first three rows and columns will be attained.

By induction, if the above process is continued on each column and row (j=2,3,4,....v-1) exactly v-j times, all of the possible (v-1)! permutations of the initial connection matrix are obtained. See example 3-2 for the generation of all permutations for a five node graph.

Example 3-2 Generation of all connection matrices for a five node graph (M_0=12345)

j=2

M_0		M_2		M_{22}		M_{222}		
12345	→	14352	→	14523	→	15234	→	(12345)

j=3

M_0		M_3		M_{33}		
12345	→	12453	→	12534	→	(12345)
M_2		M_{23}		M_{233}		
13452	→	13524	→	13245	→	(13452)
M_{22}		M_{223}		M_{2233}		
14523	→	14235	→	14352	→	(14523)
M_{222}		M_{2223}		M_{22233}		
15234	→	15342	→	15423	→	(15234)

j=4

M_0		M_4		
12345	→	12354	→	(12345)
M_3		M_{34}		
12453	→	12435	→	(12453)
M_{33}		M_{334}		
12534	→	12543	→	(12534)
M_2		M_{24}		
13452	→	13425	→	(13452)
M_{23}		M_{234}		
13524	→	13542	→	(13524)
M_{233}		M_{2334}		
13245	→	13254	→	(13245)
M_{22}		M_{224}		
14523	→	14532	→	(14523)
M_{223}		M_{2234}		
14235	→	14253	→	(14235)
M_{2233}		M_{22334}		
14352	→	14325	→	(14352)
M_{222}		M_{2224}		
15234	→	15243	→	(15234)
M_{2223}		M_{22234}		
15342	→	15324	→	(15342)
M_{22233}		M_{222334}		
15423	→	15432	→	(15423)

(XXXXX)=duplication

Total matrices = 24

(v-1)! = 24

The procedure used to generate all of the T-triangles will differ only slightly from the above process and it will utilize the results obtained in that

(a) each column may be used successively only v-j times, and

(b) for any T-triangle T_{jk}, $j \leq k$.

3.4 GENERATION OF ORIENTED T-TRIANGLES

In proceeding from connection matrix transformation through column and row translations to T-triangle transformation through column and row translations, it should be noted that row i of the T-triangle is equal to row i-1 of the connection matrix. Therefore when translating column j of the T-triangle, the i=j-1 row must also be translated.

An oriented T-triangle, T_j, can be generated from an original oriented T-triangle, T_0, by the following steps.

(a) Move all entries of the i^{th} row (i=j-1) of T_0 directly down to the bottom row.

(b) Shift all entries, which are located directly below the i^{th} row, upward one location, and

(c) shift all entries, which are located to the right of column j (j=i+1) upward one location and toward the left by one location, and

(d) the entries in column j are multiplied by a negative one (-1), and translated downward to the bottom row and to the extreme right of the T-triangle bottom row.

(e) All entries above the i^{th} row are left unchanged.

The above steps relate to a column j and row i=j translation on a connection matrix M_0 (corresponds to T_0) to produce a new connection matrix M_j (corresponds to T_j). By analyzing figure 3-5 it is seen that the (-1) multiplication of the translated column in step (d) above is due to the negative symmetry of the two triangles in the connection matrix. The elements in row i of the connection matrix are equal to the elements in column j except for opposite sign. By translating row i to the bottom row is therefore the same as translating column j=i around and to the bottom row and multiplying each term by (-1). The (-1) multiplying term in part (d) above therefore regains the proper sense of the elements in the new T-triangle bottom row.

In the generation of all trees, however, step (a) of the above procedure is not necessary and is in fact not desired as it will only generate duplications of trees that have already been produced from other T-triangles. The proof of this will follow the algorithm for direct tree finding.

3.5 ALGORITHM FOR T-TRIANGLE, DIRECT TREE FINDING METHOD

Algorithm

The procedure for direct tree finding by the T-triangle method is given below.

(1) Obtain an original oriented T-triangle, T_0, from a given oriented graph.

(2) Take any column j of T_0 as an "operating column" provided that

  (a) the entries of that column are not all zero, or

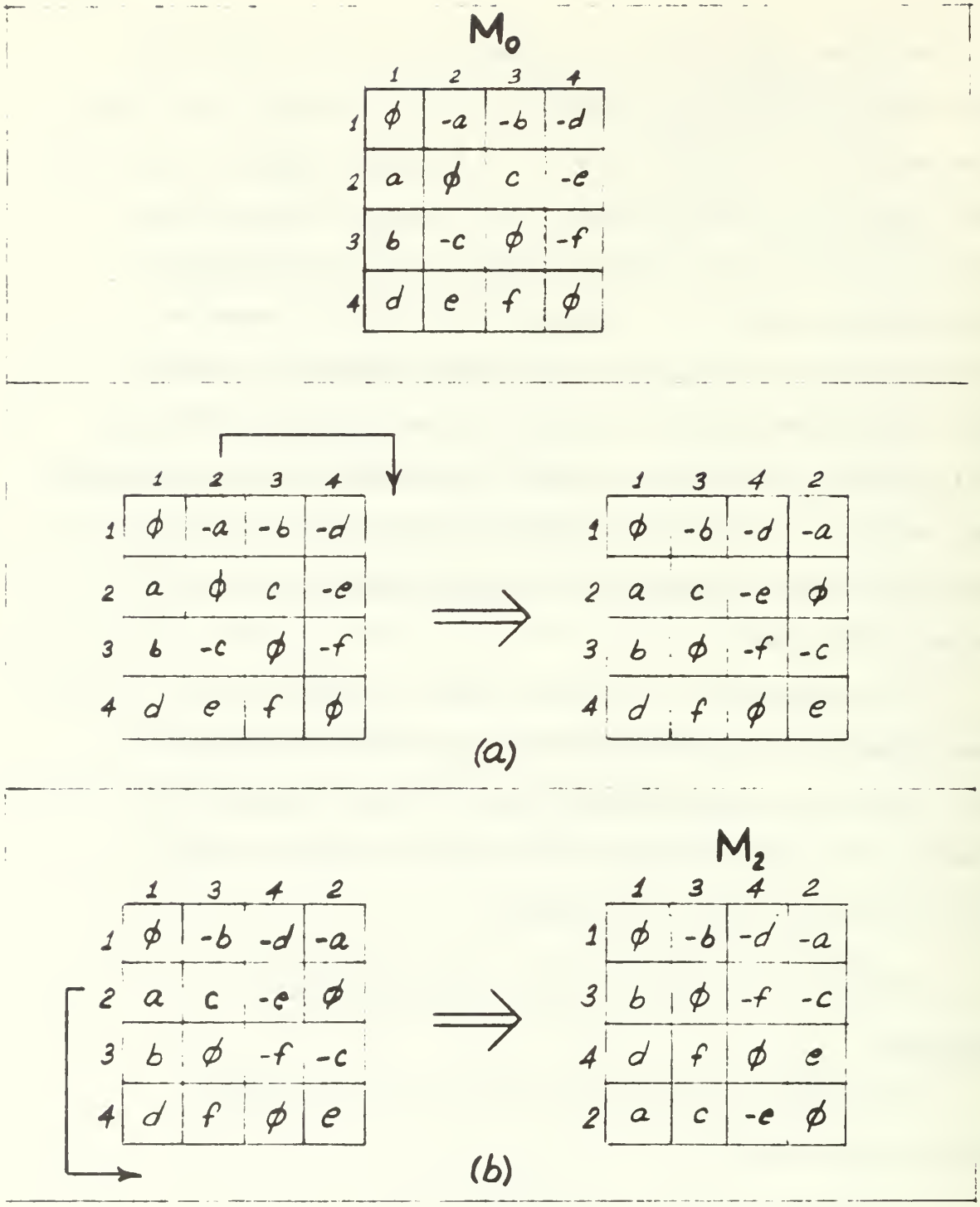


Figure 3-5 Generation of M_2 and M_0.

To generate M_2 from M_0: (a) translate column 2 to the end and (b) translate row 2 to the bottom.

(b) the entries to the left of column j and below (j-1) are not all zero.

(3) Obtain a new T-triangle, T_j, from T_0 by

(a) removing the elements of the $(j-1)^{th}$ row of T_0, and

(b) shifting all the entries, which are located directly below the $(j-1)^{th}$ row, upward one location, and

(c) shift all the entries which are located to the right of column j, upward one location and toward the left by one location, and

(d) the entries in column j are

(i) multiplied by a negative one, and

(ii) translated downward to the bottom row, and

(iii) translated to the extreme right of the T-triangle bottom row.

(4) Repeat (2) and (3) for every j=2,3,...., v-1, where v is the number of vertices in the graph.

(5) From each of the newly obtained T-triangles (call them T_j, j=2,3,..., v-1) from steps (3) and (4), generate new T-triangles (call them T_{jk}, k=j+1,..., v-1) by taking an operating column k each time as in steps (2) and (3).

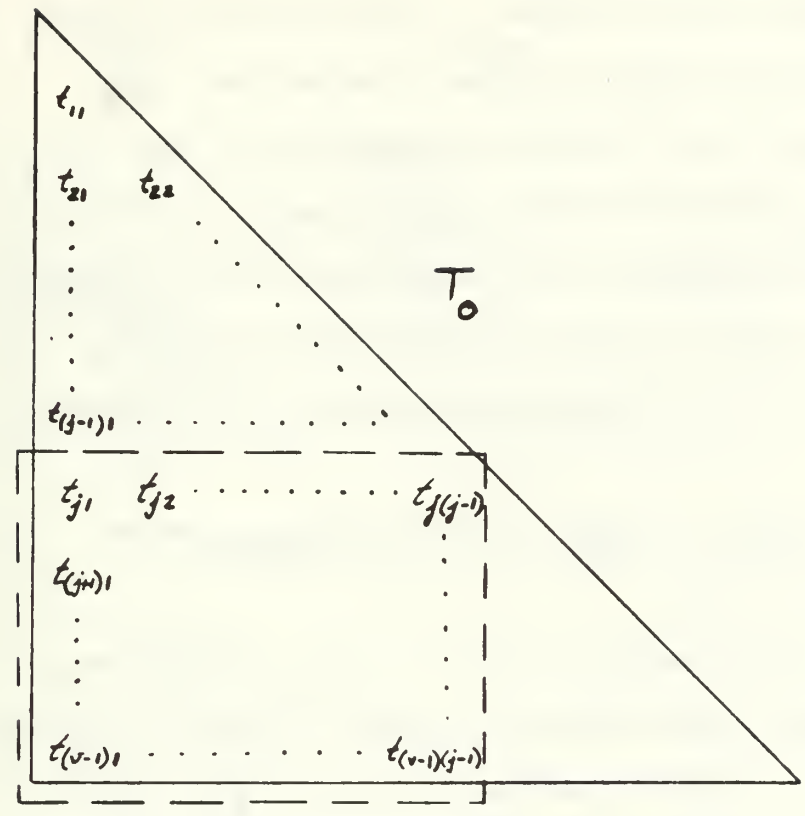
(6) Generate all T-triangles successively by repeating (5) on each newly generated T-triangle provided that the same j-column is not used for more than v-j times successively.

(7) List all the trees from each T-triangle by obtaining all possible combinations, taking one element at a time from each row of a T-triangle.

(8) Direct trees will be those trees with all elements positive.

3.6 PROOF OF METHOD

When generating a T-triangle, T_j, from a given T-triangle, T_0, by the method given in section 3.4, the elements of the i^{th} row (i=j-1) of T_0 $[t_{i1}, t_{i2}, \cdots, t_{i(j-1)}]$ would be translated to the bottom row and would become elements $t_{(v-1)1}$, $t_{(v-1)2}$, $\cdots$ $t_{(v-1)\ (j-1)}$ of T_j. However all of the trees in T_j that use these elements $t_{(v-1)\ 1}$, $t_{(v-1)2}$, $\cdots$ $t_{(v-1)\ (j-1)}$ have been previously determined in T_0. This is because columns 1,2, ..., j-1 of T_0 and T_j still have the same elements even though their orders have been modified (column k elements of T_0 = column k elements of T_j) and all trees of T_j that use the elements in the columns 1, 2, ..., j-1 have been previously determined in T_0. Figure 3-6 illustrates this relationship. Also since any T-triangle $T_{jk} \mid k \leq j$ that are derived from T_j will also have these same elements in their respective columns and no new unique trees would be obtained by using the elements of the i^{th} row of T_0. The i^{th} row (i=j-1) of T_0 should therefore be deleted when formulating T_j. If this procedure is continued throughout all of the (v-1)! T-triangle generations, all of the trees will be produced without any duplications because in each T-triangle generation only the elements that can produce new unique combinations are retained.



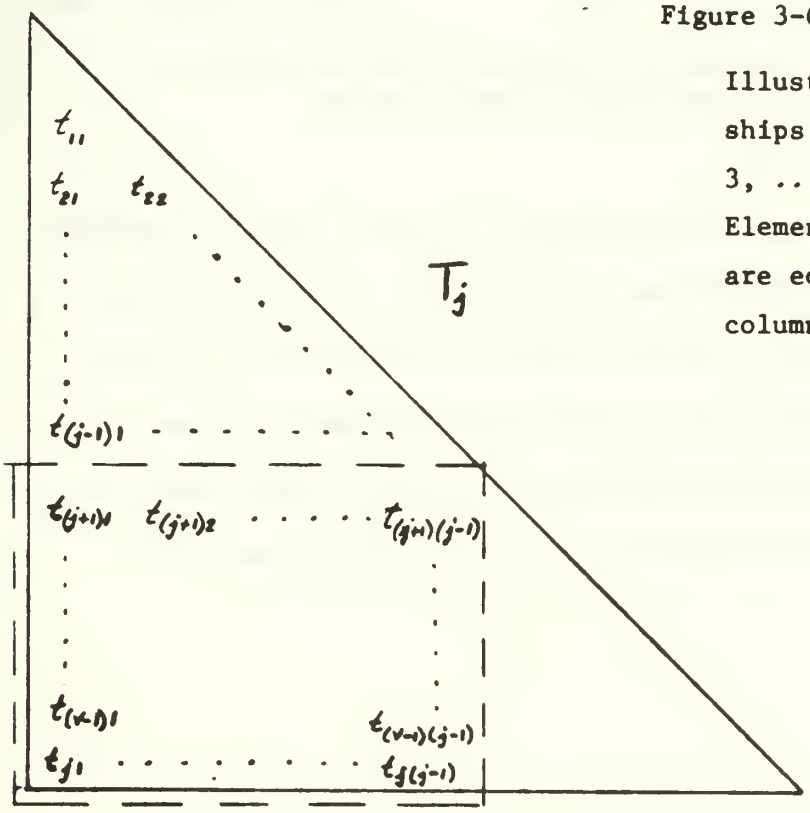


Figure 3-6

Illustration of relationships between columns 1,2, 3, ... , (j-1) of T_0 and T_j. Elements in column k of T_0 are equal to the elements in column k of T_j (k<j).

If, in generating a new T-triangle, T_j, from a given T-triangle, T_0, it is found that the elements in column j are all zero, T_j can be deleted from the (v-1)! possible T-triangles as it will have all zero elements in its bottom row and this represents a disconnected subgraph with an isolated node. No trees are therefore formed by this disconnected subgraph and all further permutations on T_j will only result in continued disconnected subgraphs with isolated nodes.

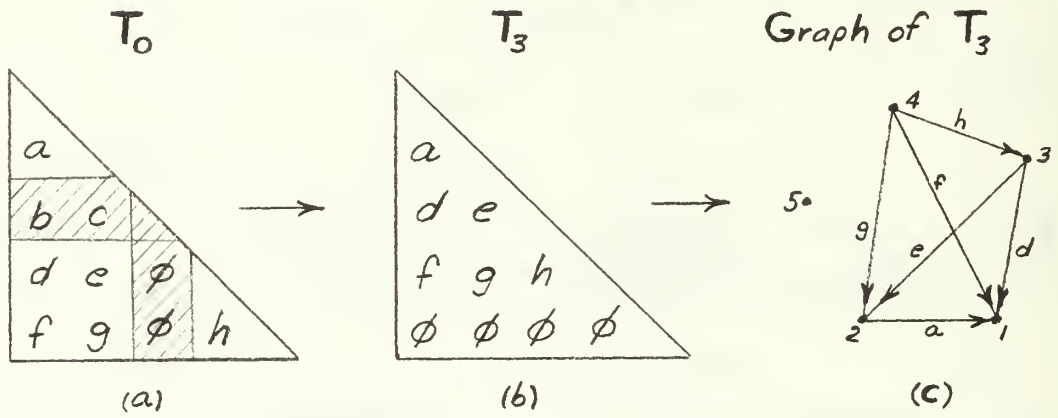


Figure 3-8 Disconnected subgraph formed by column j of T_0 having all elements zero.

If, in generating a new T-triangle, T_j, from a given T-triangle, T_0, it is found that the entries to the left of column j and below row (j-1) are all zero, T_j can be deleted from the (v-1)! possible T-triangles as it represents a disconnected subgraph with row i= j-1 an empty row. Further permutations of T_j will only result in continued disconnected subgraphs and no new unique trees can be generated.

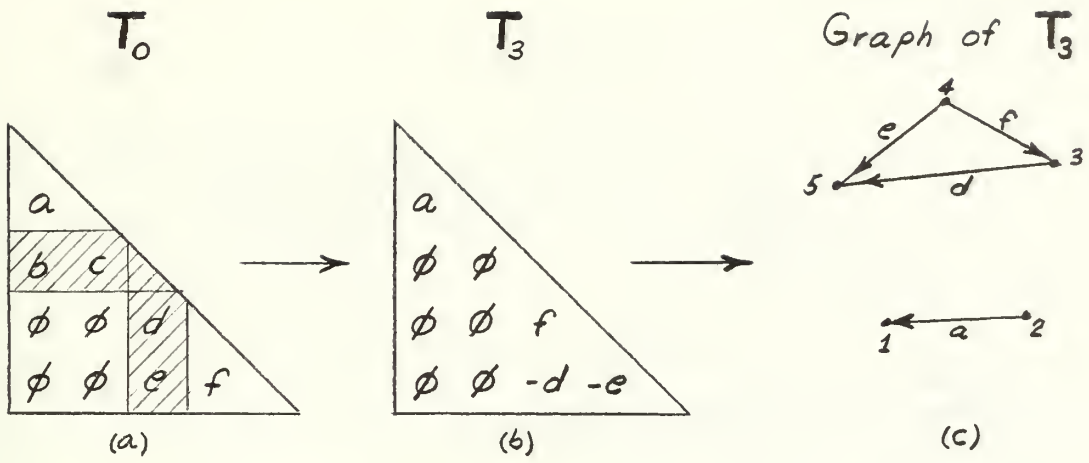
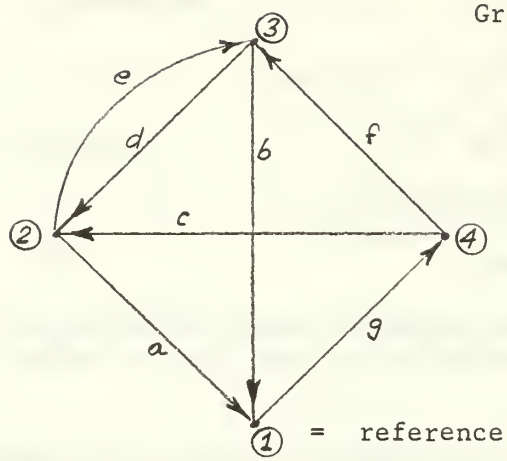
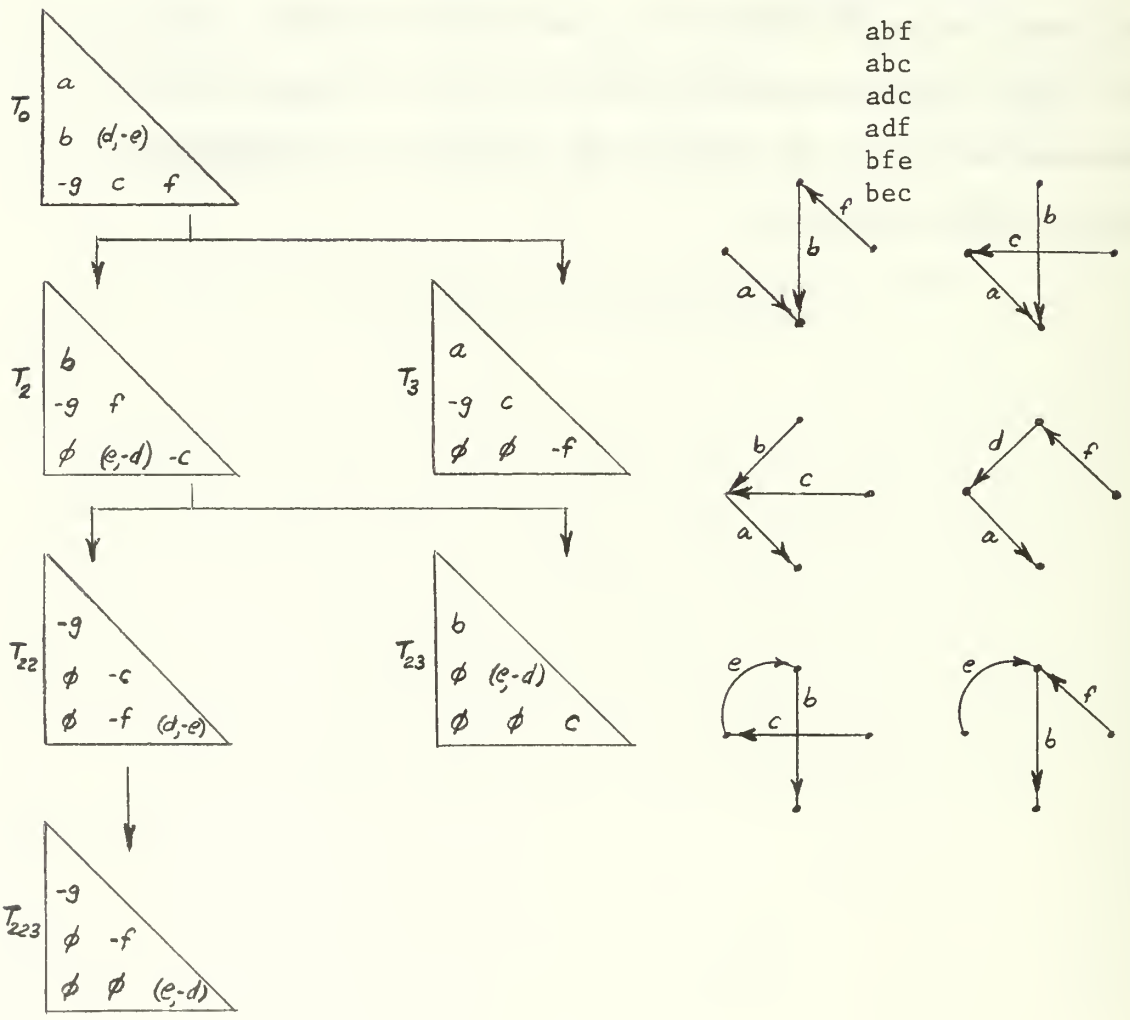


Figure 3-9 Disconnected subgraph formed by having all zero elements below row j-1 and left of column j.

If steps (1) through (6) are performed as given in the algorithm until each column has been used successively its maximum allowed v-j times, all of the desired possible (v-1)! T-triangles will be generated and all of the trees of the circuit will be generated without any duplications.

3.7 EXAMPLE OF METHOD





4. CONCLUSION

The T-triangle method for finding all of the direct trees of a given oriented graph has the following properties.

(1) Simplicity

(2) Suitable for hand calculations

(3) Suitable for computer calculations

(4) All of the trees are found without any duplications

(5) Minimal steps are required

This method should be ideally suited for computer implementation because of the minimum storage and time requirements.

BIBLIOGRAPHY

1. Wen-Tao Chang, "Determination of Network Functions By Topological Method," United States Naval Postgraduate School Thesis, June 1968.

2. Shu-Gar Chan and Wen-Tao Chang, "A Fast Tree-Finding Method," Eleventh Midwest Symposium on Circuit Theory, pp. 457-462, May 1968.

3. Wai-Kai Chen, "On the Realization of Directed Trees and Directed 2-Trees," IEEE Transactions on Circuit Theory, June 1966.

4. Wai-Kai Chen, "Topological Analysis for Active Networks," IEEE Transactions on Circuit Theory, March 1965.

5. Seshu and M.B. Reed, "Linear Graphs and Electrical Networks," Addison Wesley, 1961.

6. Seshu and N. Balabanian, "Linear Network Analysis," John Wiley and Sons, 1959.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library Naval Postgraduate School Monterey, California 93940	2
3.	Professor S.G. Chan Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	5
4.	Lt Carl Edward Willman 888 Pacific Street Monterey, California	2

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1 ORIGINATING ACTIVITY (Corporate author)

Naval Postgraduate School
Monterey, California 93940

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3 REPORT TITLE

Determination of Direct Trees By T-triangle Method

4 DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis

5 AUTHOR(S) (First name, middle initial, last name)

Willman, Carl Edward, Lieutenant, USN

6 REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
April 1969	34	6

8a. CONTRACT OR GRANT NO	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10 DISTRIBUTION STATEMENT

Distribution of this document is unlimited

11 SUPPLEMENTARY NOTES	12 SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13 ABSTRACT

A method for determining all of the direct trees of an oriented graph is presented. This method, the T-triangle method, is suitable for either hand calculation or computer implementation. The method is simple, contains a minimal amount of steps, and generates all of the direct trees without any duplications.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
direct trees						
connection matrix						
network topology						